

## Exemplo do Trabalho de Códigos

A seguir descreve-se um exemplo, com um dos códigos propostos (neste caso o código CMI), de como se devem portar os programas a realizar.

Assim, irão ser descritos os vários comandos introduzidos e os resultados obtidos, em ambiente MATLAB (representados neste texto a itálico), intercalados com comentários.

Assim, primeiramente usa-se a função *help* do MATLAB para saber quais os parâmetros necessários para executar o programa que efectua a codificação do vector binário, ou seja

```
>> help codificador
```

```
[sinal] = codificador(vector, tb, amplitude, resolucao)
```

*Implementa um codificador do tipo CMI; vector representa o conjunto de simbolos binarios a serem codificados; tb e a duracao de cada simbolo (em ms); amplitude e o modulo da amplitude de cada simbolo correspondente ao bit 1 (em V); resolucao e o numero de pontos por cada unidade de tempo; sinal representa o sinal (continuo) devidamente codificado*

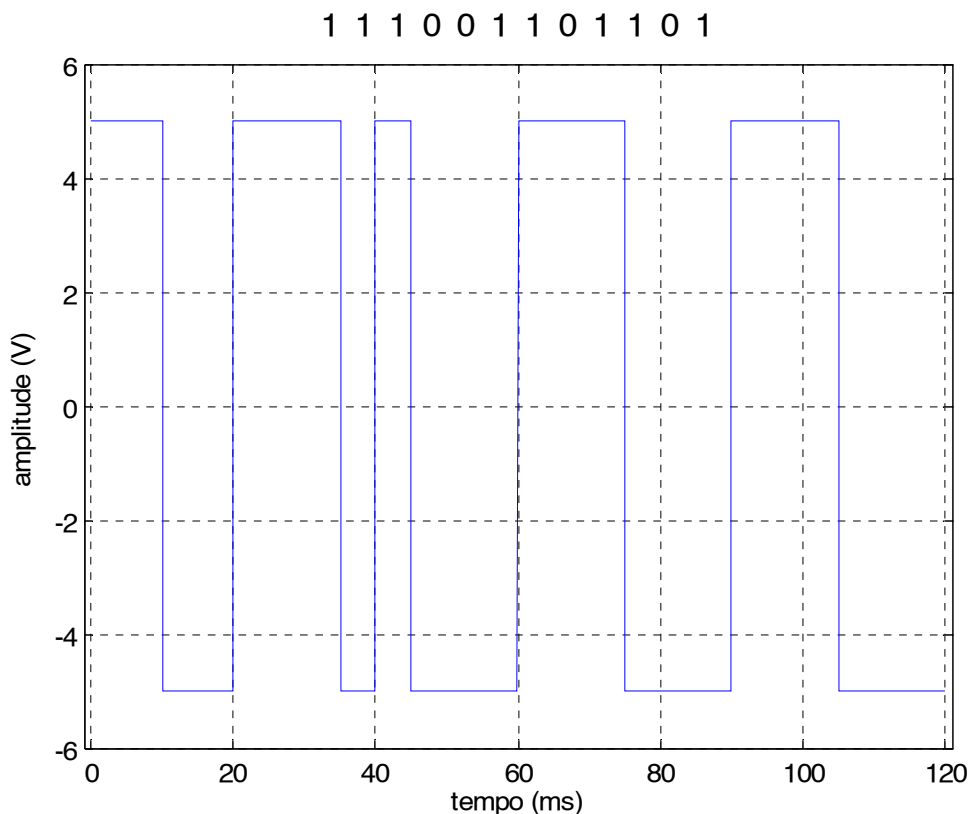
Atendendo ao descrito na ajuda, especifica-se um vector aleatório de 0's e 1's, um valor especificando a duração de cada símbolo, a amplitude que se pretende para o bit 1 e a resolução (número de pontos) pretendida para cada símbolo. Ou seja, executam-se, por exemplo, os seguintes comandos:

```
>> v = [1 1 1 0 0 1 1 0 1 1 0 1];  
>> tb = 10;  
>> amp = 5;  
>> resolucao = 100;
```

De seguida executa-se o programa com estes valores, isto é,

```
>> sinal = codificador(v, tb, amp, resolucao);
```

Neste ponto deve abrir-se uma janela com a visualização gráfica do sinal codificado gerado, vindo, neste caso,



Nesta janela, podem ver-se os seguintes pontos:

- Em cima aparece o vector que está a ser codificado;
- Como existem 12 elementos no vector com uma duração de 10 ms cada, a escala dos tempos vem de 0 a 120 (12 \* 10);
- Os valores máximo e mínimo apresentados na escala das amplitudes estão, em valor absoluto, um pouco acima do módulo da amplitude para se ter uma melhor percepção do sinal gerado.

Quanto ao decodificador, começa-se por executar a sua ajuda para saber quais os parâmetros a passar, isto é,

```
>> help decodificador
```

```
[vector_descodificado] = decodificador(sinal, tb, resolucao)
```

*Implementa um decodificador do tipo CMI; sinal representa o sinal continuo contendo o sinal CMI codificado; tb representa o tempo de duracao de cada simbolo; resolucao representa o numero de pontos por unidade de tempo; vector\_descodificado o vector decodificado*

Por último, executa-se o programa com os parâmetros do sinal obtido anteriormente, e com os mesmos parâmetros para a duração e a resolução para cada símbolo, ficando

```
>> vec_desc = descodificador(sinal, tb, resolucao)
```

```
vec_desc =
```

```
1 1 1 0 0 1 1 0 1 1 0 1
```

O vector *vec\_desc* corresponde ao sinal CMI descodificado e está de acordo com o original.

Comentários finais:

- Este texto corresponde somente a um exemplo, podendo adaptar quantos e quais os parâmetros a passar a cada programa, dependendo do trabalho de cada grupo;
- Os nomes a dar aos programas deverão ser: *codificador\_CURSO\_GRUPO* e *descodificador\_CURSO\_GRUPO*, em que *CURSO* deve ser *ee* ou *ei*, conforme o curso (Engenharia Electrotécnica ou Engenharia Informática) a que pertença o grupo e *GRUPO* deve ser o número do grupo (por exemplo, o programa que efectua a codificação do código atribuído ao grupo I do curso de Engenharia Electrotécnica deve chamar-se *codificador\_ee\_1*);
- A forma de representação do sinal codificado e do vector descodificado são meramente ilustrativos, podendo cada trabalho desenvolver os meios que entenderem necessários para melhor representar a informação que entenderem relevante;
- A inclusão de uma ajuda (*help*) em cada programa é obrigatória, de modo a que se saiba o que o mesmo faz e os parâmetros necessários.

Além da elaboração destes programas, em ambiente MATLAB, cada grupo deverá realizar um relatório relativamente a este trabalho, nos moldes discutidos nas aulas, nomeadamente com uma introdução sobre o código atribuído, a metodologia adoptada na concepção e elaboração dos programas, ilustrações de aplicação, resultados obtidos e sua avaliação crítica, conclusões e referências. De referir que estes tópicos são apenas orientações, ficando a elaboração dos relatórios, e sua respectiva valorização, a cargo de cada grupo.

Bom trabalho!

ESTiG, 21 de Dezembro de 2004,

O Docente,  
Filipe Santos Moreira